Extracting characteristics of a product from product description

Извлекување карактеристики на производ од описот на производот

Студент
Дејан Симоновски

Ментор
проф. Д-р Димитар Трајанов

Скопје, 2024

The goal of this project is to extract the characteristics of a product from its description.

Method 1: Regex

We have used two methods, which we will soon illustrate below. First of which, is regex. This method uses a common pattern to distinguish characteristics manually. The results from this method have proven to be very effective, even if the dataset isn't the most consistent, all the features that cannot be separated to its own key, are added to the 'Features' key. For example,

Преносен звучник Моќност:15W Mikrofon AUX /USB/TF/Bluetooth RGB-LED Батерија:3.7V1200mAh

This would usually be

{'Производ': 'Преносен звучник ',

'Моќност': '15W '

'Влез': 'Mikrofon,AUX,/USB/TF/Bluetooth'

'Осветлување': 'RGB-LED'

'Батерија': '3.7V1200mAh'

But since our regex matcher can only do so much, we have the following output: {'Моќност': '15W',

'Батерија': '3.7V1200mAh',

'Features': 'Преносен звучник Mikrofon AUX /USB/TF/Bluetooth RGB-LED'}

As visible from the example, all non-key values are added to the 'Features' key. Because of this, if a key is not found in this row, we search for its value in features exclusively.

Method 2 Cohere AI:

In this method, we send our rows to the AI via API requests with a similar query to this:

Extract the key-value pairs from the following product specification text. If the key or value is missing, still infer the correct pairs. Identify additional information under the category 'Features'.

Example 1:

Input:

"Screen Size: 6.5 inches

Resolution: 1080x2400 pixels

….

….

Now, extract the key-value pairs for the following product specification text:

{text}

And we send our subsample (because it is a large dataset, just for this example)

After this processing is over, we ask the user what he wants via prompt. We process this prompt with Cohere AI once more, for example:

Query: Show me televisions with Bluetooth and 1080p resolution, and we get the following in JSON format:

```
{
  "Connection": "Bluetooth",
  "Resolution": "1080p"
}
```

We proceed with translating the keys, Connection to Поврзување and Resolution to Резолуција via ‘Translator’ plugin → from translate import Translator

We have our prompt in JSON, we have our characteristics in JSON. Now it’s just a matter of doing a search and showing the compatible products.

We will mostly focus on ‘extraction’ rather than searching, because that is the topic at hand at this moment. If we take the first row from each sample, we will see which method is more accurate of the two:

Regex:

```
{'Моќност': '15W', 'Батерија': '3.7V1200mAh', 'Features': 'Преносен звучник Mikrofon AUX /USB/TF/Bluetooth RGB-LED'}
```

Cohere AI:

```
{'Product': 'Преносен звучник', 'Power': '15W', 'Connectivity': 'AUX, USB, TF, Bluetooth', 'Lighting': 'RGB-LED', 'Battery': '3.7V, 1200mAh', 'Features': 'Built-in microphone'}
```

Conclusion:

The added benefit from using Cohere in this case, is that Cohere automatically translates the keys, and adds keys where needed, if it’s just ‘Bluetooth’ it will make it ‘Connection’:’Bluetooth’, Regex on the other hand, throws everything unknown in the ‘Features’ key. Cohere also does this, but to a lesser extent.

Obviously, using Cohere will always give better results, unless our dataset is consistent, which is unlikely knowing that humans enter data, and humans are prone to making mistakes, unlike machines.

Code available at Google drive